

Upgrading From Liferay DXP 7.1 to 7.2

Table of Contents

The DXP Upgrade Process	1
Breaking Changes to Liferay DXP 7.2	1
Upgrade Tooling	1
Upgrade Checklist	1
Back up and Make Copies of Everything	1
Modules	2
Headless APIs	3
Themes	3
Document Repository	3
Search	4
Infrastructure	4
Database	5
Fix Packs and Patching	5
Running the Upgrade	6
Upgrade Steps	6
Moving Forward	10

The DXP Upgrade Process

If you have upgraded from prior versions of Liferay DXP, you will know that upgrades can be a time- and resource-intensive project. However, in Liferay DXP 7.1, we introduced several features that make things easier. Customers who have recently upgraded from 6.2 to 7.0 can be assured that these changes will make the task of upgrading easier than upgrades in the past.

This paper is a high-level overview of the effort involved in moving from Liferay DXP 7.x to 7.2. For an in-depth look at the process, please review the Liferay documentation on help.liferay.com.

Breaking Changes to Liferay DXP 7.2

We have started documenting the breaking changes in new versions of Liferay from the version before. These changes might affect the functionality of certain modules. For more information about API changes that could affect developed modules, please review the [Breaking Changes list](#). This list is updated regularly.

Upgrade Tooling

One of the new features introduced in Liferay DXP 7.1 and available in Liferay DXP 7.2 is the ability to restart an upgrade. In prior versions, if an upgrade failed, the database had to be restored and the whole process would begin again. When 7.0 was introduced, module upgrades could be restarted upon failure, but an upgrade failure on the core database required a restoration from a backup. With the new upgrade tool, there is no need to restore from a backup. Simply address the error that stopped the upgrade and resume it.

Upgrade Checklist

The following is a brief overview of the necessary steps required prior to starting an upgrade. These are the main points of preparation and general tasks that should be done to ensure a successful upgrade.

Back up and Make Copies of Everything

Before getting started, it's highly recommended that you back up and make copies of the data and anything deployed to the application server. For a Liferay project, that is usually the database, document repository and application server. For some application servers, a backup of the entire app server can be made (Tomcat, Wildfly, etc.) whereas other app servers cannot be copied in full.

The backup process preserves any deployed applications and plugins, all data and all documents in the Liferay system in a manner that would allow for a quick restoration of the data if needed for any reason.

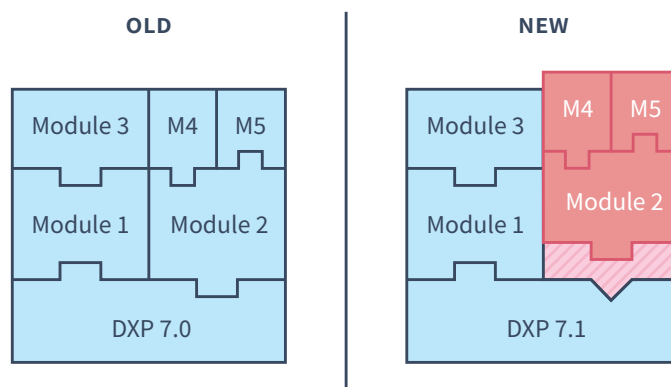
Modules

The good news about upgrading from Liferay DXP 7.x to 7.2 is that the big hump is behind you. There were massive architecture changes from 6.2 to 7.0, as Liferay Portal moved from the monolithic approach to the much more modular, OSGi approach of Liferay DXP. This required a rewriting of almost all portlets and plugins to convert to modules.

If you have already done the conversion to modules, please cross-reference any custom modules with the [Breaking Changes list](#) to ensure they continue to work properly.

If you have existing modules in 7.0 or 7.1 deployed, they can be separated into two kinds of modules: Liferay Marketplace Apps and build modules. If you are using any Marketplace apps, check the Liferay Marketplace for an updated version of the app. Modules that you have built will need to be at the very least recompiled against a Liferay DXP 7.2 workspace for it to function. Changes in Liferay APIs may cause modules to cease to function properly, so don't forget to consult the Breaking Changes list linked above.

Liferay now provides a tool to check if module versions have changed between versions of Liferay DXP. This is useful if a custom module is making use of a Liferay module and to ensure that things still work across versions. You can access the tool [here](#).



Above: In Liferay DXP 7.0, all the modules fit together and work fine, but in 7.2, one of the extension points (APIs) has changed, so if taking Module 2 as is from 7.0 and deploying it into 7.2 causes it to break, then any subsequent modules that depend on it, such as Module 4 and Module 5, will also break.

Headless APIs

New in Liferay DXP 7.2 is a commitment to a fully headless system, using APIs written using the OpenAPI standard. Moving forward APIs should not break between versions.

Themes

In Liferay DXP 7.0, themes were built with Bootstrap 3. The 7.1 release moved to Bootstrap 4. In an effort to keep backwards compatibility, we have introduced a Compatibility Layer for Bootstrap. Themes written with Bootstrap 3 will attempt to be compatible with Bootstrap 4, so you won't have to rewrite the whole theme. However, the Compatibility Layer may not be perfect, so you may experience some issues.

Liferay DXP 7.2 continues using Bootstrap 4, the same as 7.1. If you are coming from 7.1 to 7.2, you will not need to migrate Bootstrap versions. However, you may need to update your module requirements and metadata. As with custom modules, these should be recompiled against a 7.2 workspace.

With the change to Bootstrap 4, you may prefer to rewrite your theme. The Compatibility Layer exists so that you don't have to do this immediately and gives you options to apply fixes so your theme is compatible with Bootstrap 4 until you are ready to rewrite.

For more information about the Bootstrap compatibility layer, please [review the documentation on Help Center](#).

Document Repository

While there are no major changes to the document repository from Liferay DXP 7.x to 7.2, you should always make sure you backup the document repository file system and double check that the path to the repository is correct. By default, Liferay DXP stores the file system in `[Liferay Home]/data/document_library` of each DXP bundle or installation. When upgrading, you can copy the old document repository to the new location, leaving the old bundle to function as a rollback point.

If you are using a Store type other than `FileSystemStore`, don't forget to make the change in `portal-ext.properties` in the new Liferay DXP deployment.

Search

With Liferay DXP 7.0, we introduced Liferay Enterprise Search, powered by Elasticsearch. Elasticsearch is known for their rapid release cadence, and frequently retire old versions and release new versions. If you are using Elasticsearch, and you're not on a supported version, please take the time to update your Elasticsearch deployments to supported versions.

If you're using Solr, the Compatibility Matrix lists Solr 7.x as the supported version.

Liferay DXP	DXP 7.0	DXP 7.1	DXP 7.2
Search	Solr 5.2-5.5, Elasticsearch 2.2-2.4	Solr 7.x, Elasticsearch 6.1.x	Solr 7.x, Elasticsearch 6.1.x-6.5.x

If you are using Liferay Enterprise Search, don't forget to deploy the connector to interface Liferay with your search infrastructure.

While not required for Liferay DXP to function with Elasticsearch, the Enterprise Search connector does connect Liferay Search APIs to Elasticsearch. Without the connector, certain specific search features may not work or may resort to a less efficient method, such as database search.

Infrastructure

Given that there is a new version of Liferay DXP, your company may be considering changing your infrastructure. New servers? Going to cloud? Going on-premise?

Liferay DXP 7.2 has an updated Compatibility Matrix that outlines the supported versions of various Application Servers, Databases and Browsers. If you're coming from 7.x and going to 7.2, the supported versions won't have significant changes and there may be some overlap.

Sample setup showing supported versions in 7.0 vs 7.1 vs 7.2. Note that older versions in 7.1 are not supported.

Liferay DXP	DXP 7.0	DXP 7.1	DXP 7.2
OS	CentOS 6, CentOS 7	CentOS 7	CentOS 7
Application Server	Tomcat 8.0, Tomcat 8.5	Tomcat 9.0	Tomcat 9.0
Database	MySQL 5.6, MySQL 5.7	MySQL 5.7	MySQL 5.7, MySQL 8

You can find more information on supported versions and technologies in the [updated compatibility matrix](#).

Database

For the database, the main task is to back it up and create a duplicate database by restoring the backup. This tests the backup and restore mechanism in case it is ever needed in regards to live data. It also creates a duplicate database that can be tested on without using a live data set.

Fix Packs and Patching

It is generally recommended to resolve and patch all issues prior to upgrading. You don't want issues to snowball as part of the upgrade process, so having issues resolved prior to beginning your upgrade is very important. If you have no active issues, updating to the latest fix pack on both the old version of Liferay DXP and the new version is recommended. Fix packs in Liferay DXP contain mechanisms on verifying data and ensuring data integrity, which is beneficial when preparing for an upgrade between versions.

[Download the latest Liferay DXP 7.0 fix pack >](#)

[Download the latest Liferay DXP 7.1 fix pack >](#)

[Download the latest Liferay DXP 7.2 fix pack >](#)

As of this writing, the latest available fix pack for DXP 7.0 was Fix Pack 85, the latest available fix pack for DXP 7.1 was Fix Pack 12 and the latest available fix pack for DXP 7.2 was Fix Pack 1.

	DXP 7.0	DXP 7.1	DXP 7.2
Latest Fixpack (Aug 2019)	Fix Pack 85	Fix Pack 13	Fix Pack 1

Running the Upgrade

One of the major features introduced in Liferay DXP 7.0 was the Upgrade Tool. This continues to be available in 7.2, but with a few added benefits. Not only is the ability to run the upgrade module by module still present, but now the upgrade tool will allow you to continue an upgrade that has failed. This saves time and energy without having to restore backups every time the upgrade failed.

Upgrade Steps

1. Set Indexing to Read Only

Indexing while upgrading leads to performance penalties and is an unnecessary use of system resources that could be devoted to completing the upgrade faster. To set the index to read only, an OSGi config file will have to be created.

1. Create a file called

```
com.liferay.portal.search.configuration.  
IndexStatusManagerConfiguration.config
```

2. Place the file in [Liferay Home]/osgi/config

3. Edit the file and add in

```
indexReadOnly="true"
```

2. Find Upgrade Script

The upgrade tool is located in [Liferay Home]/tools/portal-tools-db-upgrade-client and contains all the necessary files for starting the upgrade.

3. Upgrade Process Tuning Options

One of the key benefits of the upgrade tool is the ability to set its own JVM parameters and tuning. The regular day-to-day usage of the system does not reflect the same usage as the upgrade process, so they should be tuned differently.

Regular Usage of Database	Upgrade Usage of Database
95% read (SELECT)	98% write (UPDATE, DELETE, INSERT)
5% write (UPDATE, DELETE, INSERT)	2% read (SELECT)

In the upgrade tool directory, there is a `db_upgrade.sh` and `db_upgrade.bat` file. These are the files that will be used. By default, the upgrade tool has the following parameters:

```
-Dfile.encoding=UTF8 -Duser.country=US
-Duser.language=en -Duser.timezone=GMT -Xmx2048m
```

In order to change those parameters, one could modify the `.sh` or `.bat` file and run the file. The option for specifying parameters is also possible through the “-j” flag. An example of setting the JVM to 10GB at runtime would be:

```
db_upgrade.sh -j “-Dfile.encoding=UTF8
-Duser.country=US -Duser.language=en
-Duser.timezone=GMT -Xmx10240m”
```

For `.bat` file usage, simply replace the `.sh` file extension.

4. Run Upgrade Script

To run the upgrade, simply run the script `db_upgrade.sh` or `db_upgrade.bat` depending on the OS. On first run, the upgrade process will ask a series of questions about the type of database and the locations of various files depending on the application server type.

The upgrade tool will ask a series of questions on first run to establish the location of various directories and database related information for the bundle. The tool will offer default answers; you can hit “Enter” to accept these default answers without typing out the entire string.

```

D:\liferay\demo\liferay-7.1.10-fs\tools\portal-tools-db-upgrade-client>db_upgrade.bat
[ jboss jonas resin tcserver tomcat weblogic websphere wildfly ]
Please enter your application server (tomcat):
tomcat
Please enter your application server directory (D:\liferay\demo\liferay-7.1.10-fs\tomcat-9.0.6):

Please enter your extra library directories in application server directory (/bin):
/bin
Please enter your global library directory in application server directory (/lib):
/lib
Please enter your portal directory in application server directory (/webapps/ROOT):
/webapps/ROOT
[ db2 mariadb mysql oracle postgresql sqlserver sybase ]
Please enter your database (mysql):
mysql
Please enter your database JDBC driver class name (com.mysql.jdbc.Driver):
com.mysql.jdbc.Driver
Please enter your database JDBC driver protocol (jdbc:mysql://):
jdbc:mysql://
Please enter your database host (localhost):
localhost
Please enter your database port (none):

Please enter your database name (/lportal):
/fsv2upgrade
Please enter your database username:
lruser
Please enter your database password:
*****
Please enter your Liferay home (D:\liferay\demo\liferay-7.1.10-fs):

```

5. Post Upgrade-Script Tasks

a. Portal-ext.properties

Every new version of Liferay DXP comes with a few new properties settings that are defaults in the older version. It's highly recommended to review the new default settings and compare with existing settings to ensure compatibility. These settings are located inside the Liferay deployment such as [Liferay WAR File]/WEB-INF/classes/portal-legacy-x.y.properties where x.y is the current version of Liferay DXP you are on. If you have upgraded across multiple versions, you will need the older versions.

Additionally, some portal-ext.properties settings may have been moved to OSGi config file settings, so certain settings will no longer be used and have been moved. For more information on config files, please refer to the [documentation on understanding system configuration files](#).

b. Verify Your Data

Once the upgrade process is complete and properties have been verified, it's a good idea to start the system to make sure everything is there. If you haven't deployed any custom modules or themes, those may not show up, or in the case of the theme, it may revert to the classic Liferay theme. This may be a good time to test out core Liferay DXP functionality and

ensure things are working the way you expect. A basic check would be to ensure any content is present, pages are laid out the way they should be and permissions and users are set up correctly.

6. Deploy Modules and Themes

Don't forget to deploy any custom modules or themes to the upgraded system. Once they have been deployed, check again that things are working the way you expect. Don't forget that any custom modules that were built need to be at least recompiled for DXP 7.2, as noted above in the checklist.

Other things like Liferay Marketplace Apps can also be deployed at this time as well, such as the connectors for SharePoint and Elasticsearch integration that Liferay provides.

Why not check everything at once? Why do two checks? Two checks allows for isolation of variables. If you run into issues, it will be easier to identify if the issue is caused by the upgrade process, an issue in core Liferay DXP 7.2 or customizations. When everything is tested in one pool, it can be difficult to isolate the root cause.

7. Tune Application Server JVM and Other Performance Items

Running Liferay DXP with default JVM settings is possible, but not reflective of any real-world usage. In normal usage, a vast majority of the work is read statements on the database, with occasional write/updates; however, during the upgrade, there are few read statements and the majority of the operations are update/write. Performance tuning is dependent on many variables, such as the technology stack itself, the number of concurrent users, the number of concurrent users at peak time and the types of operations that users will be using. This process is an iterative process, usually involving changing a setting, testing that and then making further adjustments.

Please refer to the [Liferay DXP 7.2 Deployment Checklist](#) for details on performance tuning and getting ready for going live in production.

8. Stand up a Standalone Elasticsearch or Solr Server

In Liferay DXP 7.x, the default search engine is a stripped down, embedded version of Elasticsearch. For a production environment, it is strongly recommended to replace the embedded engine with a standalone Elasticsearch or Solr setup. As noted above, the supported search providers in Liferay DXP 7.2 are Elasticsearch 6.1.x-6.5.x and Solr 7.x.

Once the server has been setup, connect it into Liferay, deploy the necessary connectors and reindex.

For more information, read the [documentation on installing a search engine](#).

9. Verify With Stakeholders

At this point, all the difficult work should be done, and it's time to verify that everything is as it should be. Talk to your various stakeholders and testers to ensure that all data and functionality is present, that the look and feel is correct and that the upgraded version meets all branding and style guidelines.

For an in-depth look at upgrading to Liferay DXP 7.2, please [refer to the documentation](#).

Moving Forward

Our Liferay Global Services team is ready to provide a deep analysis of your specific requirements, help you form a personalized upgrade plan and offer you inside knowledge on setting your company up for success with Liferay. Learn more about Liferay DXP 7.2 and the consulting services available to you by contacting sales@liferay.com or visiting liferay.com/consulting.



Liferay makes software that helps companies create digital experiences on web, mobile and connected devices. Our platform is open source, which makes it more reliable, innovative and secure. We try to leave a positive mark on the world through business and technology. Hundreds of organizations in financial services, healthcare, government, insurance, retail, manufacturing and multiple other industries use Liferay. Visit us at liferay.com.

© 2019 Liferay, Inc. All rights reserved.